

Phutball is PSPACE-hard

Dariusz Dereniowski

Department of Algorithms and System Modeling,
Gdańsk University of Technology, Poland
deren@eti.pg.gda.pl

March 20, 2009

Abstract: We consider the $n \times n$ game of Phutball. It is shown that, given an arbitrary position of stones on the board, it is a PSPACE-hard problem to determine whether the specified player can win the game, regardless of the opponent's choices made during the game.

Keywords: computational complexity, games, Phutball, Pspace hardness

1 Introduction

There is a deep mathematical theory developed for analyzing combinatorial games [1, 8]. The researchers work on the algorithmic techniques which are useful for finding good game strategies for many board games, including Phutball [4, 5]. The paper [6] introduces the notion of generalized threads and this technique is used to solve some Go instances and the author suggest that this approach could be effective for other board games, like Phutball. T.Cazenave used an approach called Gradual Abstract Proof Search to show that 11×11 Phutball is a win for the first player [7]. The game is *loopy*, i.e. it is possible to obtain a configuration of stones which already appeared in one of the previous turns – some combinatorial aspects of loopy games were considered in [24]. In this paper we are interested in the complexity of the game rather than in manipulating and analyzing the rooted tree describing the game. Several generalizations of one-player games turns out

to be NP-complete: Peg Solitaire [26], Minesweeper (the problem of testing consistency) [20], Same Game [3]. However, most of the board games (especially two-player games) appear to be harder: Checkers [23], Hex (a generalization to graphs) [13], Othello (Reversi) [19], Sokoban [9], Go [21, 22, 27], Dyson Telescopes [12], Rush Hour [14] or Amazons [16].

The Phutball [2] game is usually played on a 19×19 Go board. Initially a black stone is placed in the middle of the board. The players make their moves alternately. A player makes his move by either placing a white stone in an unoccupied position, or makes a sequence of *jumps* over horizontal, vertical or diagonal sequences of white stones. Each jump is performed by moving the black stone, called *ball*, over a line of white stones (no empty space between the ball and the line is allowed if we want to make a jump) and placing the ball on the board on the first unoccupied position after the last white stone in the line. The white stones are removed from the board immediately after the jump. Each player tries to move the ball on or over the opponent's *goal line*. The goal lines are two opposite edges of the board. We consider a natural generalization with an arbitrary size of the board and initially a black stone placed in the middle of the board.

As indicated in [2], Phutball is not the kind of game where you can expect a complete analysis. The authors in [17] considered a simplified version of the game, i.e. the case where there is only one dimension and it turns out that according to the presented examples, the one-dimensional version still seems to be hard to analyze. Moreover, given an arbitrary position in the 2-dimensional Phutball game, it is an NP-complete problem to determine whether the current player can win the game in his next move [11]. However, as indicated in several papers [7, 10, 11, 18], the complexity of the Phutball game is still open. In this paper we place the problem of determining whether the current player has a winning strategy in the class of PSPACE-hard problems.

2 A graph game

We start this section by describing the rules of a game played on a graph. Then we prove that this game is PSPACE-hard. The graph constructed on the basis of a problem known to be PSPACE-complete is defined in such a way that its topology allows to code it as a configuration of stones in the Phutball game.

The game described in the following is played on a directed graph. For completeness we list here some basic definitions. A directed graph G is a pair $G = (V(G), E(G))$ with a vertex set $V(G)$ and a set of directed edges $E(G)$ (each $e \in$

$E(G)$ is an ordered pair of two vertices). We say that H is a *subgraph* of G , $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A *directed path* $P = (\{v_1, \dots, v_n\}, E(P))$ from v_1 to v_n is a graph with edge set $E(P) = \{(v_i, v_{i+1}) : i = 1, \dots, n-1\}$. The vertices $V(P) \setminus \{v_1, v_n\}$ are the *internal vertices* of P .

The input of the game is a directed graph $G = (V(G), E(G))$, a set $C \subseteq V(G)$, a vertex $s \in C$, and a relation $R \subseteq V(G) \times E(G)$ between the vertices and the edges of G . If $(v, e) \in R$ then we say that a vertex v is *pointing* an edge e . Denote by $R^{-1}(E(G))$ the set of vertices v for which there exists $e \in E(G)$, such that $(v, e) \in R$. The players of the game will be called \exists -player and \forall -player. We will also use a notation that if a symbol X refers to one of the players then \bar{X} is the other player.

At each point of the game there is a unique *active vertex*. The players must follow the rules:

Rule 1 (*initialization*). The \exists -player starts the game. Initially s is the active vertex.

Rule 2 (*a move*). Let $u \in C$ be the active vertex. The current player X selects a vertex $v \in C \cup R^{-1}(E(G))$ and a directed path $P \subseteq G$ from u to v such that all internal vertices of P are in $V(G) \setminus (C \cup R^{-1}(E(G)))$. The edges of P are removed from G , v becomes the active vertex, and \bar{X} becomes the current player. We say that X *moves from u to v* .

Rule 3 (*game end*). If the current player cannot make a move, i.e. there is no directed path P from the active vertex to a vertex $v \in C \cup R^{-1}(E(G))$, then the current player loses the game. If the current player moves from u to $v \in R^{-1}(E(G))$ then he wins the game.

Let us recall the PSPACE-complete Quantified Boolean Formula (QBF) problem [25]. Given a formula Q in the form

$$Q_1 x_1 \cdots Q_n x_n F(x_1, \dots, x_n),$$

decide whether the formula is true, where $Q_i \in \{\exists, \forall\}$ for $i = 1, \dots, n$. In our case we use a restricted case of this problem where $Q_1 = \exists$, $Q_{i+1} \neq Q_i$ for $i = 1, \dots, n-1$, the integer n is even, and F is a 3CNF formula, i.e. $F = F_1 \wedge F_2 \wedge \cdots \wedge F_m$, where $F_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ and each literal $l_{i,j}$ is a variable or the negation of a variable, $i = 1, \dots, m$, $j = 1, 2, 3$.

Given Q , we create a directed graph G . For each variable x_i define the corresponding variable component $G(x_i)$:

$$V(G(x_i)) = \{a_i, b_i, c_i, d_i, e_i, f_i, g_i\},$$

$$E(G(x_i)) = \{(a_i, b_i), (a_i, c_i), (b_i, e_i), (c_i, f_i), (e_i, d_i), (f_i, d_i), (d_i, g_i)\},$$

for $i = 1, \dots, n$. We connect the variable components in such a way that $(g_i, a_{i+1}) \in E(G)$ for each $i = 1, \dots, n-1$.

Then we define the formula component $G(F)$:

$$V(G(F)) = \{x_i, y_i, z_i : i = 1, \dots, m\} \cup \{w_{i,j} : i = 1, \dots, m, j = 1, 2, 3\},$$

$$E(G(F)) = \{(x_i, y_i), (y_i, z_i), (z_i, w_{i,1}), (w_{i,1}, w_{i,2}), (w_{i,2}, w_{i,3}) : i = 1, \dots, m\} \cup \{(x_i, x_{i+1}) : i = 1, \dots, m-1\}.$$

Fig. 1(a) shows the formula component while Fig. 1(b) gives the variable component. To finish the construction of G let $(g_n, x_1) \in E(G)$ and introduce a vertex g_0

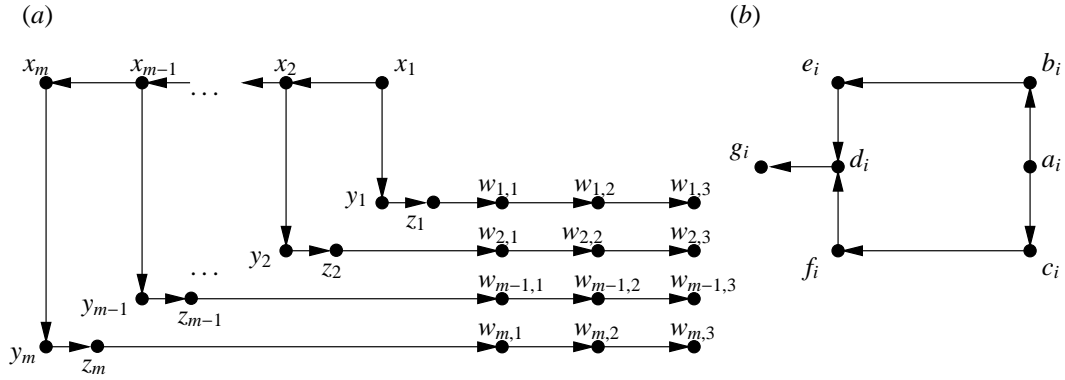


Figure 1: The graphs (a) $G(F)$ and (b) $G(x_i)$

connected to the graph in such a way that $(g_0, a_1) \in E(G)$.

The input to our graph game is the directed graph G defined above, the set $C = \{g_0, \dots, g_{n-1}\} \cup \{z_1, \dots, z_m\}$, $s = g_0$ and R containing a pair $(w_{i,j}, (b_l, e_l))$ (respectively $(w_{i,j}, (c_l, f_l))$) for $i \in \{1, \dots, m\}$, $j \in \{1, 2, 3\}$, $l \in \{1, \dots, n\}$, iff $l_{i,j} = x_l$ ($l_{i,j} = \bar{x}_l$, resp.). Observe that initially $R^{-1}(E(G))$ contains all the vertices $w_{i,j}$, $i = 1, \dots, m$, $j = 1, 2, 3$, because a vertex $w_{i,j}$ corresponds to the literal $l_{i,j}$, which equals x_l or \bar{x}_l for some $l \in \{1, \dots, n\}$. However, during the game the set R^{-1} gets smaller due to the fact that some of the edges of G are removed from G .

All the edges of the graph have vertical and horizontal orientations (as shown in Figs 1 and 2) and the lines of stones in the Phutball game corresponding to the edges of the graph will preserve this topology.

Let us consider the following complete example of our reduction. Given a formula Q

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4), \quad (1)$$

Fig. 2 depicts the corresponding graph G . The dashed arcs represent the elements of the relation R , the vertices in C are denoted as white nodes, while the vertices in $V(G) \setminus C$ are the black nodes. Note that for each vertex $w_{i,j}$ there is exactly one element $(w_{i,j}, e) \in R$. Since the \exists -player starts the game and the ac-

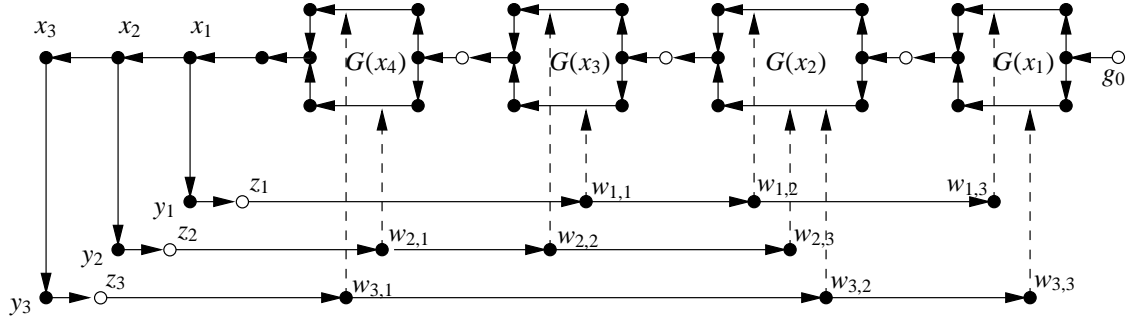


Figure 2: A complete instance of the graph G corresponding to the formula in (1)

tive vertex is g_0 , two moves are possible, i.e. $V(P)$ contains $g_0, a_1, b_1, e_1, d_1, g_1$ or $g_0, a_1, c_1, f_1, d_1, g_1$. After the move $w_{1,3}$ or $w_{3,3}$ does not belong to the set $R^{-1}(E(G))$, respectively.

The game obtained in the reduction has a special structure, which makes it quite easy to analyze. Here we list three straightforward facts describing the structure of the game.

Fact 1 If the active vertex is g_{i-1} , $i \in \{1, \dots, n\}$ then the current player makes a move from g_{i-1} to g_i and the directed path P removed from G contains one of the following sequences of edges:

$$(g_{i-1}, a_i), (a_i, b_i), (b_i, e_i), (e_i, d_i), (d_i, g_i), \quad (2)$$

$$(g_{i-1}, a_i), (a_i, c_i), (c_i, f_i), (f_i, d_i), (d_i, g_i). \quad (3)$$

Furthermore, the \exists -player makes such a move for $i = 1, 3, 5, \dots, n-1$ while the \forall -player makes this move for $i = 2, 4, 6, \dots, n$ (recall that n is even). \square

Fact 2 Let g_n be the active vertex. The \forall -player is the current player and he makes a move from g_n to a vertex z_i and the path P contains the edges

$$(g_n, x_1), (x_1, x_2), \dots, (x_{i-1}, x_i), (x_i, y_i), (y_i, z_i), \quad (4)$$

where $i \in \{1, \dots, m\}$. \square

Fact 3 Assume that z_i , $i \in \{1, \dots, m\}$, is the current vertex. The \exists -player is the current player. If there exists such a vertex $w_{i,j}$ that $(w_{i,j}, e) \in R$ and $e \in E(G)$ then \exists -player wins the game. Otherwise he cannot make a move and he loses the game. \square

Theorem 1 *The above graph game is PSPACE-hard.*

Proof: We show that the \exists -player has a winning strategy if and only if the corresponding quantified formula Q is true. If we write $e \in E(G)$ then we mean that e is an edge of G at the current stage of the game.

First, assume that the formula is true. Define the strategy as follows. If the active vertex is g_{i-1} and the variable x_i , $i \in \{1, 3, 5, \dots, n-1\}$ is true (in an assignment of Boolean values to the variables forcing F to be true) then the \exists -player traverses the sequence of edges as stated in (3). Otherwise the \exists -player traverses the edges listed in (2). By Fact 2, the \forall -player chooses in his last move an index $i \in \{1, \dots, m\}$ and ends the move at a vertex z_i . Since F is true, F_i is true, and consequently, there exists a literal $l_{i,j}$, $j \in \{1, 2, 3\}$, which is true. If $l_{i,j} = x_l$ for some $l \in \{1, \dots, n\}$, then x_l is true which implies that $(b_l, e_l) \in E(G)$. Moreover, by the definition of R , $w_{i,j}$ is pointing (b_l, e_l) . Similarly, if $l_{i,j} = \overline{x_l}$ for $l \in \{1, \dots, n\}$, then x_l is false, so $(c_l, f_l) \in E(G)$ and $(w_{i,j}, (c_l, f_l)) \in R$. So, for each choice of i by the \forall -player there exists a vertex $w_{i,j}$ pointing an edge of G . So, the \exists -player has a win.

Assume now that \exists -player has a winning strategy. We prove that the formula Q is true. If the \exists -player traverses the edges in (2) in order to reach g_i then define x_i to be false, otherwise let the value of x_i be set to true. Then the \forall -player chooses any of the paths (2) or (3) which corresponds to setting an arbitrary Boolean value to the variable x_{i+1} quantified by \forall . When the Boolean values have been assigned to the variables then, by Fact 2, the \forall -player chooses a vertex z_i . Since the \exists -player has a winning strategy, by Fact 3, at least one of the vertices $w_{i,j}$, $j \in \{1, 2, 3\}$, is pointing an edge which still belongs to G . By Fact 3, the literal $l_{i,j}$ of F_i is true. Since i has been chosen arbitrarily, the formula F is true. \square

3 Transformation of G to the Phutball game

In the following we transform the input to the graph game, i.e. a directed graph G , a set $C \subseteq V$, a starting vertex s and a relation R , defined in Section 2, into a configuration of stones of the Phutball game. Note that we do not give a reduction between the two problems, but we only show how to code a well structured instances of the graph game. This, together with Theorem 1, will give a desired reduction from the QBF problem to the Phutball game. For brevity we will use the symbols from the previous section used to denote the vertices of G to refer to the points on the board (see e.g. Fig. 3(a)). Only the vertices in C will be coded using special gadgets. Because of the direct correspondence between the vertices of G and the fields on the Phutball board we will use the labels used for the vertices to denote the fields. It will be clear from the context whether we refer to a vertex or to a point on the board.

Let the upper (respectively lower) edge of the board be the \forall -player's (\exists -player's, resp.) goal line. The vertices in $V(G) \setminus C$ are coded as the empty points on the board. We will choose those empty points in such a way that if there is an edge $(u, v) \in E(G)$ then the points corresponding to u and v will have the same horizontal or vertical coordinates. The edges of the graph correspond to the (horizontal or vertical) sequences of stones. The starting vertex is also coded as an empty spot and it initially contains the ball. The configuration of stones corresponding to the variable component $G(x_i)$ for $i = 1, 3, 5, \dots, n-1$ is given in Fig. 3(a) while the Fig. 3(b) gives the variable component for $i = 2, 4, 6, \dots, n$. In all the figures of this section, the dots ending a vertical line of stones indicate

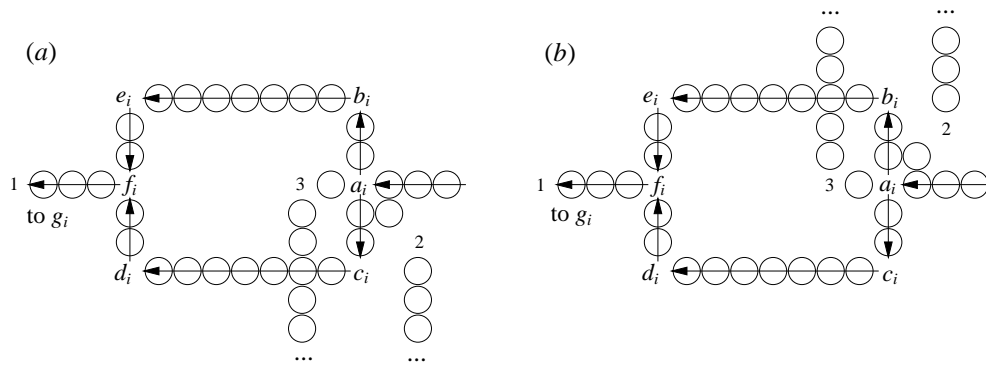


Figure 3: The variable component $G_F(x_i)$ for (a) $i = 1, 3, 5, \dots, n-1$, and (b) $i = 2, 4, 6, \dots, n-2$

that the line ends at the appropriate (upper or lower) goal line. We will use two types of configurations corresponding to the vertices g_i , $0 < i < n$. Fig. 4(a) (Fig. 4(b)) presents the configuration corresponding to g_i for $i = 1, 3, 5, \dots, n-1$ ($i = 2, 4, 6, \dots, n-2$, respectively). In order to make the analysis consistent we will use the label g_i , $0 < i < n$, to mark a point on the board as shown in Fig. 4. Roughly speaking, such a gadget forces the following sequence of events: one player makes a jump ending at field g_i (according to an arrow on the right hand side), then two stones are placed at points 8 and 18 (each by one player) and finally the other player makes a move of two jumps (as indicated by the second arrow) leading the ball directly to the point a_{i+1} . To obtain the configuration of stones

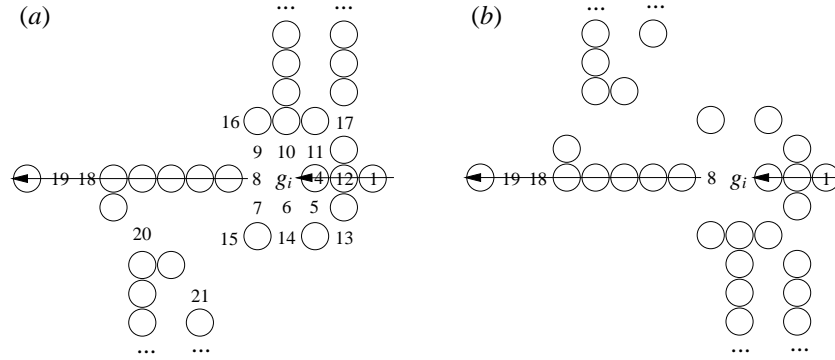


Figure 4: Configuration of white stones corresponding to vertices (a) g_i , where i is odd, (b) g_i , where i is even

corresponding to $G(x_i)$, denoted in the following by $G_P(x_i)$, for $i = 1, 3, 5, \dots, n-1$ (respectively for $i = 2, 4, 6, \dots, n-2$) we connect the gadgets in Figs 3(a) and 4(a) (Figs 3(b) and 4(b), resp.) in such a way that the points marked by 1 in both pictures refer to the same place on the board.

We will use the following correspondence between the edges of G and the lines of stones in G_P : an edge of G

$$(x, y) \in \{(g_0, a_1), (g_n, x_1)\} \cup \bigcup_{1 \leq i \leq n} E(G(x_i)) \setminus \{(g_i, a_{i+1})\}$$

corresponds to a line of white stones between the points x and y on the board, while (g_i, a_{i+1}) , $i = 1, \dots, n-1$, corresponds to two lines of stones: between g_i and 19 of $G_P(x_i)$ and between 19 of $G_P(x_i)$ and a_{i+1} of $G_P(x_{i+1})$. Note that two points in a line between g_i and 19 are by the definition unoccupied, but the game

is set up in such a way that when the ball is about to move from g_i then there is a line of white stones between g_i and 19 of $G_P(x_i)$ which we are going to prove later.

Note that we used the Facts 1-3 listed in the previous section to obtain Theorem 1. Now we prove these facts for the corresponding configurations of stones on the board. Then we may conclude that the game of Phutball simulates the graph game which will give us a desired reduction from the QBF problem. If x and y are two points on the board then $x \rightarrow y$ denotes a jump from x to y and removing all the stones between x and y (we will use this symbol in such a way that all the conditions required by the rules of the game for making a jump will be satisfied).

Proof of Fact 1: In the terms of the phutball game we are going to prove, by an induction on i , that if a ball is at g_{i-1} , $1 < i \leq n$, and is X the player making the next move then the following sequence of moves occurs:

- (i) X places a white stone at 8 of $G_P(x_{i-1})$,
- (ii) \bar{X} places a white stone at 18 of $G_P(x_{i-1})$,
- (iii) X makes a sequence of jumps over the lines of stones corresponding to the edges given in (2) or (3).

Moreover, X is the \exists -player (\forall -player) for odd (even, respectively) values of i . For $i = 1$ the situation is similar to the case when $i > 1$ except that only (iii) is done.

So, assume that the ball is at g_{i-1} . By the induction hypothesis, the white stones on the right hand side of g_{i-1} are no longer on the board. Moreover, both for odd and even values of i , the lines of stones next to 10 and 17 (20 and 21) of $G_F(x_{i-1})$ lead to the X 's (\bar{X} 's, respectively) board line. We have that X must place a white stone, because he cannot make a jump. Clearly, he cannot put a stone at 10 of $G_P(x_{i-1})$. If he does not occupy one of the fields 4-11 of $G_P(x_{i-1})$ then in the next turn \bar{X} puts a white stone at the point 10 of $G_P(x_{i-1})$ and it is easy to see that \bar{X} wins the game. Observe, that if \bar{X} is able to move the ball to the point 17 then he is one jump away from his opponent's goal line. Note that if X places a stone in one of the points 4, 5, 6, 7, 9, 11 of $G_P(x_{i-1})$ then \bar{X} can reach 17 and win the game. Thus, X places a white stone at the field 8 of $G_P(x_{i-1})$, i.e. (i) occurs. Then, \bar{X} can either: (1) make one of the moves $g_{i-1} \rightarrow 18$, $g_{i-1} \rightarrow 18 \rightarrow 20$ or $g_{i-1} \rightarrow 18 \rightarrow 20 \rightarrow 21$, but it is easy to see that in all cases his opponent wins in the next turn, or (2) put a white stone and if he chooses a field different than 18 then, similarly as in (1), he loses, because either 20 or 21 of $G_F(x_{i-1})$

is unoccupied. So, X can make jumps $g_{i-1} \rightarrow 18 \rightarrow 20$ or $g_{i-1} \rightarrow 18 \rightarrow 21$, respectively and reach his opponent's goal line in the next jump. This proves that (ii) must happen.

The fact that X must reach the point g_i by jumping over the lines of stones corresponding to (2) or (3) follows from the observation that otherwise he loses the game. In particular, if X places a white stone somewhere on the board instead of making some jumps then one of the points 3 or 2 of $G_P(x_i)$ is still unoccupied. Then, \bar{X} reaches 3 or 2 in $G_P(x_i)$, respectively, and his next jump places the ball at the X 's goal line. So, X is forced to make a sequence of jumps. If X finishes his move before reaching b_i or c_i then his moves were $g_{i-1} \rightarrow 19$, $g_{i-1} \rightarrow 19 \rightarrow a_i$, $g_{i-1} \rightarrow 19 \rightarrow a_i \rightarrow 2$ or $g_{i-1} \rightarrow 19 \rightarrow a_i \rightarrow 3$ and it is easy to see that \bar{X} wins immediately in all cases. If X reaches b_i , by jumps $g_{i-1} \rightarrow 19 \rightarrow a_i \rightarrow b_i$ (the case of c_i is analogous) then he must follow to the point g_i (by jumps $b_i \rightarrow e_i \rightarrow f_i \rightarrow g_i$) since otherwise there is a path $b_i \rightarrow e_i \rightarrow d_i \rightarrow c_i \rightarrow a_i \rightarrow 2$ which \bar{X} can follow. \square

The conversion of the formula component $G(F)$ to the configuration of stones is shown in Figs 5 and 6. In particular, Fig. 5 depicts the board representation of the edges (x_{i-1}, x_i) , (x_i, y_i) and (y_i, z_i) while Fig. 6(a) (respectively Fig. 6(b)) gives the configuration of stones coding the situation when $w_{i,t}$ is pointing an edge (b_j, e_j) ((c_j, f_j) , respectively), where $i \in \{1, \dots, m\}$, $t \in \{1, 2, 3\}$ and $j \in \{1, \dots, n\}$.

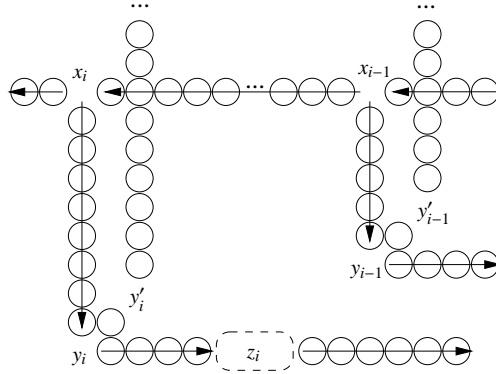


Figure 5: The configuration representing the paths in (4)

See Fig. 7 for an example where $G_P(x_2)$ is given together with $w_{1,2}$ pointing (b_2, e_2) and $w_{2,3}, w_{3,2}$ both pointing (c_2, f_2) . The configuration of stones corresponding to a vertex z_i is identical to the one in Fig. 4(a), but it is rotated with the angle of 180

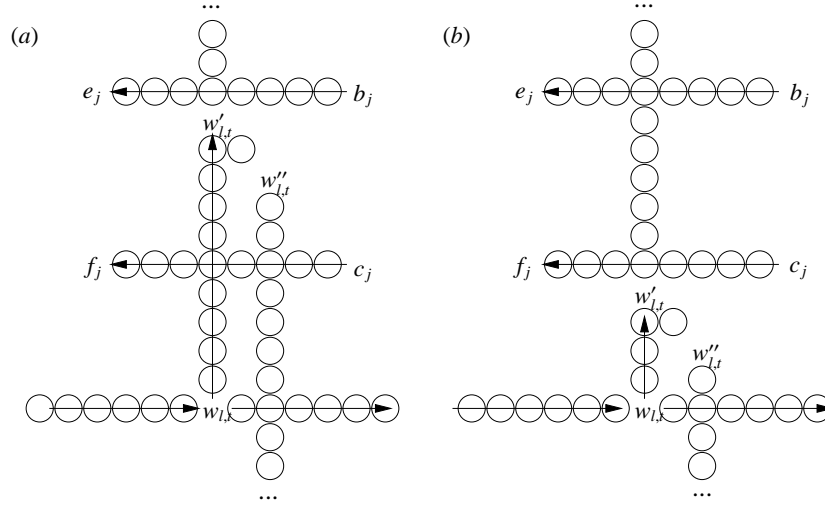


Figure 6: (a) $w_{l,2}$ and pointing to (b_j, e_j) and (b) $w_{l,3}$ and pointing to (e_i, f_i)

degrees, and we use the symbol z_i to refer to the point denoted by g_i in the case of $G_P(x_i)$'s. The configuration of stones corresponding to $G(F)$ is denoted by $G_P(F)$.

It remains to mention that the number of white stones in a line does not change the analysis of the game. In particular the distance between b_j and e_j can be arbitrary long and in our reduction it depends on the number of vertices $w_{i,t}$ pointing (b_j, e_j) or (c_j, f_j) . We require that there a distance of at least one 'field' between each pair of vertical lines appearing on the board. In this way if one stone, say from a field z , in a line between x and y has been removed from the board during the game then a jump $x \rightarrow y$, where y may be on the goal line, is replaced by two jumps $x \rightarrow z \rightarrow y$. Thanks to the distance between the vertical lines, no other jump from z is possible. If a player ends his move at z then such a situation does not differ from the case when the move would end at y and our analysis covers that.

Proof of Fact 2: From the Fact 1 it follows that when the ball is at point g_n then the \forall -player is the current player. The player cannot place a stone on the board in the forthcoming move, because then the \exists -player selects an empty point x_i (it exists, because we assumed w.l.o.g. that there are at least two variables in the formula) and

$$g_n \rightarrow x_1 \rightarrow \cdots \rightarrow x_i \rightarrow y_i \rightarrow y_i' \quad (5)$$

leads directly to the \forall -player goal line.

By a simple induction on i one can prove that when the \forall -player reaches the point x_i then he must either jump to x_{i+1} or to y_i . So, there exists $i \in \{1, \dots, m\}$ such that the \forall -player reaches y_i . He cannot end his sequence of jumps at y_i or y'_i , because, as before, he would immediately lose. By the arguments similar to the ones presented in the proof of Fact 1, the moves (i) and (ii) in the component corresponding to the vertex z_i of G must occur and the \exists -player begins his move when the ball is at z_i . \square

Proof of Fact 3: By Fact 2, the \forall -player reaches z_i . Similarly as in the proof of Fact 1 one can show that (i) and (ii), defined in the proof, occur and consequently \exists -player starts his move from z_i when the fields 8 and 18 are already occupied by white stones. The \exists -player is forced to make a jump, because otherwise the \forall -player follows a path

$$z_i \rightarrow w_{i,1} \rightarrow \dots \rightarrow w_{i,j} \rightarrow w'_{i,j} \rightarrow w''_{i,j}$$

(see Fig. 6), where $j \in \{1, 2, 3\}$ can be chosen in such a way that none of $w_{i,j}$, $w'_{i,j}$, $w''_{i,j}$ is occupied by a white stone placed by the \exists -player in his last move. So, the \exists -player has three paths to follow:

$$z_i \rightarrow w_{i,1} \rightarrow \dots \rightarrow w_{i,j} \rightarrow w'_{i,j}, \quad (6)$$

where $j \in \{1, 2, 3\}$. If he ends his move at $w'_{i,j}$ (or earlier) then the \forall -player reaches $w''_{i,j}$ and can jump to the \exists -player's board line. There are two cases to consider: (i) $w_{i,j}$ points to an edge (b_l, e_l) ; (ii) $w_{i,j}$ points to an edge (c_l, f_l) . Both cases are analogous, so we consider only (i). From the construction of the board it follows that the only continuation of moves in (6) by the \exists -player in direction different than to $w'_{i,j}$ is possible when no jump from b_l to e_l has been made during the game, and in this case the \exists -player makes a jump from $w'_{i,j}$ to the \forall -player's board line. So the \exists -player wins if and only if no jump from b_l to e_l has been made (or equivalently the edge (b_l, e_l) still belongs to the corresponding graph G). It remains to mention that the line of stones between c_l and f_l has been removed from the board, but the result is that a player makes two jumps instead of one in order to move the ball from $w_{i,j}$ to $w'_{i,j}$ or from $w'_{i,j}$ to the board line. \square

Fig. 7 presents some parts of the board obtained on the basis of the graph G corresponding to formula (1) and shown in Fig. 2.

Theorem 2 *The game of Phutball is PSPACE-hard.*

Proof: The theorem follows from Facts 1-3, the proof of Theorem 1 and an observation that the size of the board is polynomial in $n + m$. \square

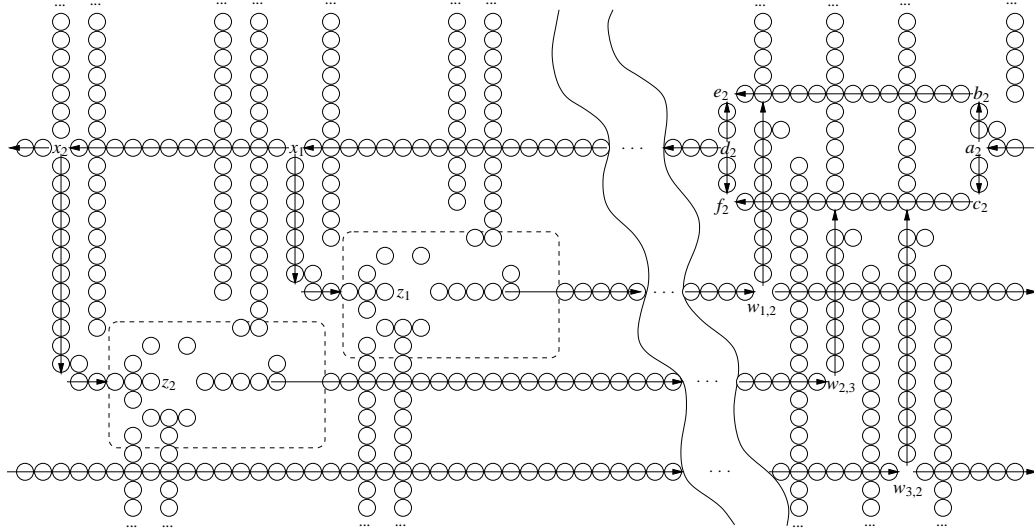


Figure 7: Some parts of the graph in Fig. 2

4 Summary

There are several natural questions one may ask about the complexity of a game. One of them is: given an arbitrary state of the game, is it possible for the current player to win in the next move? Such a problem has been considered in [11] where it has been shown that it is NP-complete for Phutball. According to the discussion in [11] the games of Checkers and Phutball have many similarities. However, it turns out that we can give a positive answer to the above question in the case of Checkers in polynomial time [11, 15]. Another question to ask about the complexity of the game is the one considered in this paper. In the case of Checkers, Fraenkel et. al. have shown that the game is PSPACE-hard. Their result has been strengthened by the paper of Robson: the game is EXPTIME-complete [23]. In this paper we developed the first result concerning the complexity of Phutball and an open question remains whether it belongs to PSPACE or is as hard as EXPTIME?

References

- [1] E. R. Berlekamp, J. H. Conway and R. K. Guy, Winning Ways for your Mathematical Plays, Vol. I, II, Academic Press, London (1982). 2nd edition: vol. 1 (2001), vol. 2 (2003), vol. 3 (2003), vol. 4 (2004), AK Peters, Natick, MA
- [2] E. Berlekamp, J.H.Conway, R.K. Guy, Philosopher's football, In: *Winning Ways for Your Mathematical Plays*, 2nd edition, A K Peters Ltd (2003) 752-755.
- [3] T.C. Biedl, E.D. Demaine, M.L. Demaine, R. Fleischer, L. Jacobson, J.I. Munro, The complexity of Clickomania, in: *More Games of No Chance* R.J. Nowakowski Ed., Cambridge University Press (2002) 389-404.
- [4] T. Cazenave, Iterative Widening, *Proceedings of IJCAI-01* **1** (2001) 523-528.
- [5] T. Cazenave, Admissible moves in two-player games, in: *Proceedings of SARA, Lecture Notes in Comp. Sci.* **2371** (2002) 52-63.
- [6] T. Cazenave, A generalized threats search algorithm, In Proceedings of the Third International Conference on Computers and Games, *Lecture Notes in Comp. Sci.* **2883** (2003) 75-87.
- [7] T. Cazenave, Gradual abstract proof search, *ICGA Journal* **25** (2002) 3-15.
- [8] J.H. Conway, *On numbers and games*, Academic Press, London (1976).
- [9] J. Culberson, Sokoban is PSPACE-complete, *Proceedings of the International Conference on Fun with Algorithms* (1998) 65-76.
- [10] E.D. Demaine, Playing games with algorithms: Algorithmic combinatorial game theory, *Proc. 26th Symposium on Mathematical Foundations in Comp. Sci., Lecture Notes in Comp. Sci.* **2136** (2001) 18-32.
- [11] E.D. Demaine, M.L. Demaine, D. Eppstein Phutball Endgames are Hard, in: *More Games of No Chance*, R. J. Nowakowski Ed., Cambridge University Press, (2002) 351-360.
- [12] E.D. Demaine, M.L. Demaine, R. Fleischer, R.A. Hearn, T. von Oertzen, The complexity of Dyson Telescopes, in: *Games of No Chance III* (to appear).

- [13] S. Even, R.E. Tarjan, A combinatorial problem which is complete in polynomial space, *Journal of the Association for Computing Machinery* **23** (1976) 710-719.
- [14] G. Flake, E. Baum, Rush Hour is PSPACE-complete, or “Why you should generously tip parking lot attendants”, *Theoretical Computer Science* **270** (2002) 895-911.
- [15] A.S. Fraenkel, M.R. Garey, D.S. Johnson, T. Schaefer, Y. Yesha, The complexity of Checkers on an $N \times N$ board, preliminary report, *Proc. 19th Annual Symposium on Foundations of Comp. Sci.* (1978) 55-64.
- [16] T. Furtak, M. Kiyomi, T. Uno, M. Buro, Generalized Amazons is PSPACE-Complete, in: *IJCAI* L.P. Kaelbling and A. Saffiotti Eds. (2005) 132-137.
- [17] J.P. Grossman, R.J. Nowakowski, One-dimensional Phutball, in: *More Games of No Chance*, R.J. Nowakowski Ed. MRSI Publications **42**, Cambridge University Press (2002) 361-367.
- [18] R.A. Hearn, Games, Puzzles, and Computation, PhD dissertation, Massachusetts Institute of Technology (2006) (to appear).
- [19] S. Iwata, T. Kasai, The Othello game on $n \times n$ board in PSPACE-complete, *Theoretical Computer Science* **123** (1994) 329-340.
- [20] R. Kaye, Minesweeper is NP-complete, *Mathematical Intelligencer* **22** (2000) 9-15.
- [21] D. Lichtenstein, M. Sipser, Go is polynomial-space hard, *Journal of the Association for Computing Machinery* **27** (1980) 393-401.
- [22] J.M. Robson, The complexity of Go, *Proceedings of the IFIP 9th World Computer Congress on Information Processing* (1983) 413-417.
- [23] J.M. Robson, N by N Checkers is EXPTIME complete, *SIAM J. Comput.* **13** (1984) 252-267.
- [24] A.N. Siegel, Loopy Games and Computation, PhD thesis, University of California at Berkeley (2005).

- [25] L.J. Stockmeyer, A.R. Meyer, Word problems requiring exponential time, *Proceedings 5th ACM Symposium on the Theory of Computing* ACM Press (1973) 1-9.
- [26] R. Uehara, S. Iwata, Generalized Hi-Q is NP-complete, *Transactions of the IEICE* **73** (1990) 270-273.
- [27] D. Wolfe, Go endgames are hard, in: *More Games of No Chance*, R.J. Nowakowski Ed. MRSI Publications **42**, Cambridge University Press (2002) 125-136.